

WHAT IS CLAIMED IS:

- 1 1. A match-and-set lock for controlling access to a resource that is shared
2 among a plurality of users N , said lock having a locked operating state and an unlocked
3 operating state controlled by a value C such that the lock is in its locked operating state
4 when $C \neq 0$ and in its unlocked operating state where $C=0$, said lock being responsive to a
5 command in the form (A, B) , said lock substituting B for C if $A=C$, whereupon:
6 the lock may be released by a user issuing the command (A, B) where $A=C$
7 and $B=0$;
8 the lock may be set by a user issuing the command (A, B) where $A=C$ and
9 $B \neq 0$; and
10 a deadlock condition may be avoided by always setting the lock to the value
11 $B=P+T*(N+1)$ where $P < (N+1)$ and identifies the user issuing the command (A, B) and T is
12 the current time stamp.
- 1 2. The match-and-set lock of claim 1, wherein the lock returns a value R , equal
2 to the lock's current content C , to an inquiring user seeking access to the resource.
- 1 3. The match-and-set lock of claim 2, wherein a return value $R=0$ denotes that
2 the resource is free, and a return value $R \neq 0$ denotes that the resource is locked by another
3 user.
- 1 4. The match-and-set lock of claim 3, wherein the return value R , when $R \neq 0$,
2 identifies the user currently locking the resource and the time when that user first locked the
3 resource.
- 1 5. The match-and-set lock of claim 4, wherein, if the inquiring user determines
2 that the user currently locking the resource has failed or restarted since locking the resource,
3 the inquiring user can gain access to the resource and reset the lock by issuing the command
4 (R, D) , where $D=P+T*(N+1)$, $P < (N+1)$ identifies the current user issuing the command and
5 T is the current time stamp.
- 1 6. The match-and-set lock of claim 1, wherein the plurality of users N are
2 processors in a multiprocessor computer system.
- 1 7. The match-and-set lock of claim 6, wherein the processors form a
2 telecommunications system.
- 1 8. A software program implemented in a multiprocessor computer system, said
2 software program controlling access to a shared resource from a plurality of processors N ,
3 by employing a match-and-set lock that has a current content C and that is responsive to a
4 match-and-set command (A, B) that writes B into the current content if $A=C$, said lock

- 5 committing the shared resource to the control of that processor that most recently set the
6 current content to a non-zero value, said software program allowing the multiprocessor
7 system to :
- 8 (a) initialize the lock by setting the current content to zero;
 - 9 (b) issue from one of the processors a command (M, S) where M equals
10 an expected current content of the lock and $S = P + T * (N + 1)$ where $P < (N + 1)$ and identifies
11 the processor issuing the command and T is a current time stamp;
 - 12 (c) if a non-zero value R is returned from the time lock, identify the
13 processor that set the lock as $P' = R \% (N + 1)$ where $R \% (N + 1)$ is the remainder of R, mod
14 (N+1), and identify the approximate locking time as $T' = R / (N + 1)$;
 - 15 (d) if processor P' is still operating and was last started before time T',
16 repeat step (b);
 - 17 (e) if processor P' is not still operating or was last started after time T',
18 issue from processor P the command (M, S) where $M = R$.
- 1 9. A method of controlling access to a shared resource from a plurality of
2 processors N, comprising the steps of:
- 3 (a) providing the resource with a locking storage function, wherein the function
4 stores either an initial value denoting that the resource is not currently held by a processor
5 or a value denoting a holding processor currently holding the resource and an initial holding
6 time when such holding began;
 - 7 (b) initializing the function by storing an initial value denoting that the resource
8 is not currently held by a processor;
 - 9 (c) querying the function to determine if the resource is currently held by a
10 processor;
 - 11 (d) if the resource is not currently held by a processor, providing to an inquiring
12 processor access to the resource and resetting the value of the function to denote the
13 inquiring processor's identity and a current time;
 - 14 (e) if the resource is currently held by a processor, determining if the holding
15 processor has failed or been restarted since taking hold of the resource;
 - 16 (f) if the holding processor has failed or been restarted since taking hold of the
17 resource, providing to the inquiring processor access to the resource and resetting the value
18 of the function to denote the inquiring processor's identity and a current time;
 - 19 (g) if the holding processor has not failed or been restarted since taking hold of
20 the resource, repeating steps (b) through (d); and

21 (h) upon release of the resource, resetting the value stored in the function to an
22 initial value denoting that the resource is not currently held by a processor.

1 10. The method of claim 9 wherein the plurality of processors N form a
2 telecommunications system.

1 11. A method of controlling access to a shared resource from a plurality of
2 processors N, said method employing a match-and-set lock that has a current content C and
3 that is responsive to a match-and-set command (A, B) that writes B into the current content
4 if A=C, said lock committing the shared resource to the control of that processor that most
5 recently set the current content to a non-zero value, said method comprising the steps of:

6 (a) initializing the lock by setting the current content to zero;

7 (b) issuing from one of the processors a command (M, S) where M
8 equals an expected current content of the lock and $S = P + T * (N + 1)$ where $P < (N + 1)$ and
9 identifies the processor issuing the command and T is a current time stamp;

10 (c) if a non-zero value R is returned from the time lock, identifying the
11 processor that set the lock as $P' = R \% (N + 1)$ where $R \% (N + 1)$ is the remainder of R, mod
12 (N+1), and identifying the approximate locking time as $T' = R / (N + 1)$;

13 (d) if processor P' is still operating and was last started before time T',
14 repeating step (b);

15 (e) if processor P' is not still operating or was last started after time T',
16 issuing from processor P the command (M, S) where $M = R$.

1 12. The method of claim 11 wherein the plurality of processors N form a
2 telecommunications system.

3